CAMBRIDGE
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/12**

Paper 1 Computer Systems **February/March 2024**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the February/March 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

PUBLISHED

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Mark scheme abbreviations

/ separates alternative words / phrases within a marking point
// separates alternative answers within a marking point
underline actual word given must be used by candidate (grammatical variants accepted)
max indicates the maximum number of marks that can be awarded
( ) the word / phrase in brackets is not required, but sets the context

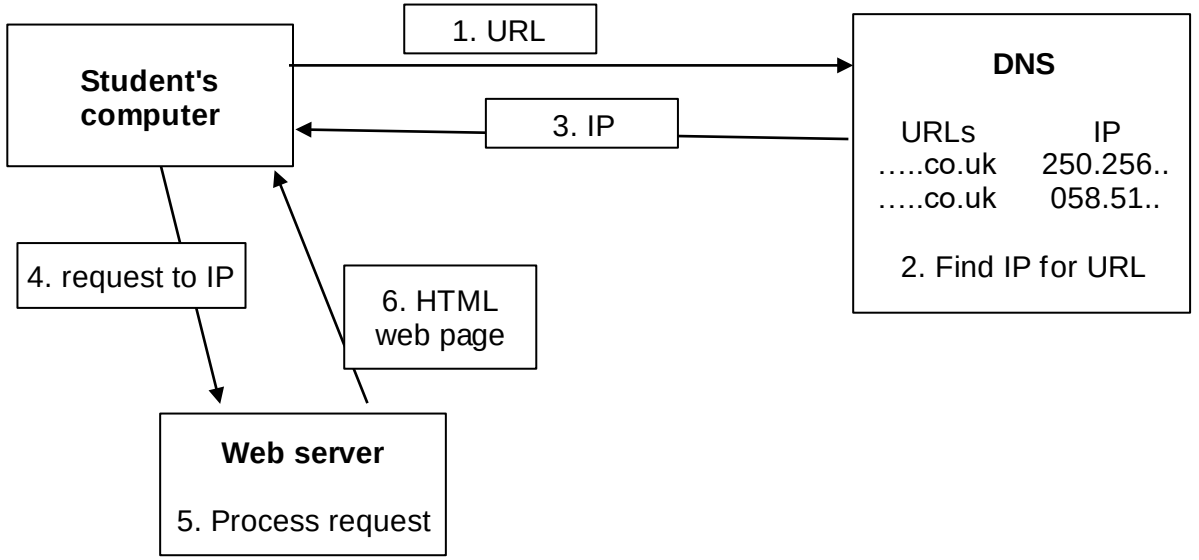
Note: No marks are awarded for using brand names of software packages or hardware.

Question	Answer	Marks
1(a)	B	**1**
1(b)(i)	A	**1**
1(b)(ii)	01001110	**1**
1(b)(iii)	• **Unique** binary/denary number given/stored for each character • The code for R is stored, then the code for E then D **in sequence**	**2**
1(c)(i)	Any **two** from: • More bits allocated to each amplitude • **Amplitudes** can be more precise • A wider range of **amplitudes** can be recorded	**2**
1(c)(ii)	Increase the sample rate	**1**

Question	Answer	Marks
2(a)	**One** mark for letter and **One** mark for matching correction. • Statement B … • …MAR stores addresses and not instructions • Statement C … • …Data is from bus not PC // Data is from address in MAR not PC	**4**
2(b)(i)	It can run 3.5 billion FE cycles **each second** // it can execute 3.5 billion instructions **each second**	**1**
2(b)(ii)	Any **two** from: • **More cores** increases/improves the performance • **More cores** mean more FDE cycles/instructions are executed each second • …because each core runs an FE cycle/instruction **simultaneously**	**2**
2(b)(iii)	Any **two** from: • **More cache** improves performance • …because **more cache** means the processor can access **more** frequently used data/instructions faster • …instead of having to access the data from the **slower-access RAM**	**2**
2(c)(i)	Any **two** from: • Volatile storage • Stores data for the processor to access directly/quickly // directly accessed by the CPU • Stores currently running data/instructions	**2**
2(c)(ii)	Any **one** from: e.g. • BIOS // bootstrap/loader • Firmware • Parts of OS	**1**
2(c)(iii)	Any **one** from: e.g. • To run programs when there is insufficient RAM to run them • To allow RAM to store more data when required	**1**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(a)</td><td>One mark each:
<table><tr><th>Function name</th><th>Description</th></tr><tr><td>managing memory</td><td>Examples:
• allocates memory to processes
• prevents two processes accessing the same memory</td></tr><tr><td>platform for running applications</td><td>allows application software to run on the computer</td></tr><tr><td>managing peripherals</td><td>Examples:
• allocates data to buffers
• transmits data to hardware
• receives data from hardware</td></tr></table></td><td>3</td></tr>
<tr><td>3(b)(i)</td><td>To indicate that something requires the attention of the processor/OS/CPU</td><td>1</td></tr>
<tr><td>3(b)(ii)</td><td>One mark for input device and matching interrupt:
e.g.
• Keyboard
Key pressed
• Mouse
Mouse moved//button clicked</td><td>1</td></tr>
</table>

Question	Answer	Marks
3(b)(iii)	Any **five** from: • Interrupt is given priority • …and placed in interrupt queue • Processor finishes current **FE cycle** for program • Processor checks interrupt priority queue // processor checks for higher priority interrupt than program/process • If lower priority processor runs next FDE cycle for program/process // if lower priority processor continues with program/process • (if higher priority) processor **stores** current process/registers on stack • Checks source of interrupt • … and calls the appropriate ISR • ISR handles/resolves interrupt • If there is another higher priority interrupt (than process) then repeat • … (otherwise) processor retrieves content of stack/registers/previous process (to continue with process from program)	**5**

Question	Answer	Marks
4	Any **five** marks for each part of diagram: • **URL** from computer/browser to DNS • DNS storing table/database of **URL** and **IPs** // DNS finding **IP** for **URL** • DNS sending **IP** to computer / browser • DNS sending to higher DNS if not found • Web browser/computer sending request to **IP** of **web server** • **Web server** processing request • Web page data sent from **server** to computer / web browser 	**5**

Question	Answer	Marks
5	**One** mark for each term in correct place: • physically • blockchains • time-stamp • traced A digital currency does not exist **physically**, it can only be accessed electronically. Some digital currencies have digital ledgers called a **blockchains**. These are decentralised databases where each transaction is stored as a new set of data with a **time-stamp** and is linked to the previous set of data. This means that transactions cannot be altered, only new transactions added, which allows the location of the data to be **traced**.	**4**

Question	Answer	Marks
6(a)	Any **one** from: • It has **electrical** components // by example • It is programmable	**1**
6(b)(i)	Any **five** from: • Sensor **continuously** sends the **digitised** value / reading / distance to the microprocessor • Microprocessor compares the data / signal to the stored data of a person and distance of 3m • If the data / signal is less than (or equal to) a person and within 3m • …a **signal** is sent to **actuator** to make the tractor **stop** / apply the brakes • If the data/signal is greater than 3m no action is taken • If stopped and data/signal is not a person and/or more than 3m a **signal** is sent to **actuator** to make the tractor **start** • The whole process repeats until turned off	**5**

Question	Answer	Marks
6(b)(ii)	**One** mark for sensor and **One** mark for matching use: e.g. • Accelerometer … • …to adjust for uneven ground // to detect if the tractor crashes • Proximity … • …to detect if near the end of the field // to detect other obstacles • Light … • … to identify when to turn the headlights on	**2**
6(c)	**Any three** from: e.g. • Set-up cost may be high • Maintenance cost may be high • … needs skilled workers/expert to fix • Farmer may need reskilling in how to use it … • ... which could be costly • Leads to deskilling of farmers/workers • Farmer may need fewer employees • … leading to unemployment • Can malfunction • … and not recognise a person and fails to stop • Changing function can be expensive	**3**
6(d)	Any **three** from • Knowledge base • Rule base • Inference engine • Interface	**3**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(e)(i)</td><td>Any three from:
• The same data is transmitted back (from computer to tractor)
• Tractor compares both sets of data
• … if they are identical there is no error // reverse
• Tractor sends confirmation of accuracy if the same // Tractor resends the data if they are different // Tractor transmits error if they are different</td><td>3</td></tr>
<tr><td>6(e)(ii)</td><td>One mark for two correct
Two marks for five correct
Three marks for all eight correct

<table>
<tr><th></th><th>parity bit</th><th>bit 7</th><th>bit 6</th><th>bit 5</th><th>bit 4</th><th>bit 3</th><th>bit 2</th><th>bit 1</th></tr>
<tr><td>byte 1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>byte 2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>byte 3</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>byte 4</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>byte 5</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>byte 6</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>byte 7</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>parity byte</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
</table></td><td>3</td></tr>
</table>

Question	Answer	Marks
7(a)(i)	If the data is intercepted, it cannot be understood	**1**
7(a)(ii)	**Four** from: • Symmetric has a shared key… • … to encrypt **and** decrypt • Both the sender and receiver know the key • Asymmetric has different keys // a public key **and** a private key • …public to encrypt the data **and** private to decrypt • …anyone can know the public key but only those intended know the private key	**4**
7(b)(i)	Any **two** from: e.g. • Destination address/IP • Sender address/IP • Packet number	**2**
7(b)(ii)	Any **one** from: • Control the route the packet takes • Send each packet towards its destination • Choose more efficient route	**1**

Question	Answer	Marks
8(a)(i)	EC	**1**
8(a)(ii)	Any **one** from: • Easier for humans to read/remember • Shorter for humans to enter • Less likely for humans to make mistakes • Easier for humans to spot errors/debug • Takes up less space **onscreen**	**1**

Question	Answer	Marks
8(b)(i)	**One** mark for working, **one** mark for answer • e.g. showing flip and add 1 10110111 = 01001001 • –73	**2**
8(b)(ii)	00101101	**1**
8(c)	**One** mark each • Divide • ...by 16 / 2^4	**2**